ФПМИ БГУ

Вычислительные методы алгебры
Лаборатоная работа 9

Нахождение собственных значений и собственных векторов методом Леверье и Фаддеева

Подготовил:
Ткачук Павел
2 курс 1 группа

Преподаватель:
Будник Анатолий Михайлович

21 декабря 2016 г.

1 Постановка задачи

Входные данные:

$$A_{\text{исх.}} = \begin{bmatrix} 0.6444 & 0.0000 & -0.1683 & 0.1184 & 0.1973 \\ -0.0395 & 0.4208 & 0.0000 & -0.0802 & 0.0263 \\ 0.0132 & -0.1184 & 0.7627 & 0.0145 & 0.0460 \\ 0.0395 & 0.0000 & -0.0960 & 0.7627 & 0.0000 \\ 0.0263 & -0.0395 & 0.1907 & -0.0158 & 0.5523 \end{bmatrix}$$

Для того чтобы все собственные значения были действительными числами, домножим слева исходную матрицу на ей транспонированную:

$$A = A_{\text{исх.}} A^T_{\text{исх.}} = \begin{bmatrix} 0.4965221 & -0.02976049 & -0.10906373 & 0.13191428 & 0.09195098 \\ -0.02976049 & 0.18575662 & -0.05029722 & -0.06272879 & -0.0018678 \\ -0.10906373 & -0.05029722 & 0.59823034 & -0.06163865 & 0.17564755 \\ 0.13191428 & -0.06272879 & -0.06163865 & 0.59248754 & -0.02931901 \\ 0.09195098 & -0.0018678 & 0.17564755 & -0.02931901 & 0.34390336 \end{bmatrix}$$

Задача:

1. Для данной матрицы A вычислить коэфициенты $p_1, p_2, \ldots, p_n$ собственного многочлена (метод Леверье, метод Фаддеева)

   $$P(A) = \lambda^n - p_1\lambda^{n-1} - p_2\lambda^{n-2} - \ldots - p_{n-1}\lambda - p_n$$

   (проверить совпадает ли $p_1 = Sp(A)$ и $p_n = \det(A)$)

2. По полученному характеристическому многочлену вычислить максимальное собственное значение λ_{max} матрицы

3. Вычислить собственный вектор $\vec{v}$ соответствующий данному значению (метод Фаддеева)

4. Найти невязку $\vec{r} = A\vec{v} - \lambda_{max}\vec{v}$

2 Алгоритм

2.1 Метод Леверье

1. Находим значения S_k

   $$S_k = Sp\, A^k,\ k = 1, \ldots n$$

2. По полученным значениям вычисляем коэфициенты характерестического многочлена

   $$p_k = \frac{1}{k}\left(S_k - \sum_{i=1}^{k-1} p_i S_{k-i}\right),\ k = 1, \ldots, n$$

2.2 Метод Фаддеева

1. Находим матрицы $A_1, A_2, \ldots, A_n$

   $$\begin{aligned} &A_1 = A, && Sp\, A_1 = q_1, && B_1 = A_1 - q_1 E, \\ &A_k = AB_{k-1}, && \frac{1}{k} Sp\, A_k = q_k, && B_k = A_k - q_k E, \quad k = 2, \ldots, n \end{aligned}$$

2. Полученные значения $(q_1, q_2, \ldots, q_n)$ являются соответсвующими коэфициентами $(p_1, p_2, \ldots, p_n)$ характеристического многочлена исходной матрицы

3. Находим λ_{max} - макисмальный корень характеристического уравнения

   $$\lambda^n - p_1\lambda^{n-1} - p_2\lambda^{n-2} - \ldots - p_{n-1}\lambda - p_n = 0$$

4. По полученному λ_{max} вычисляем собственный вектор, соотвествующий данному значению

$\vec{v} = \lambda_{max}^{n-1} e + \lambda_{max}^{n-2} b_1 + \lambda_{max}^{n-3} b_2 + \cdots + \lambda_{max} b_{n-2} + b_{n-1}$, где
$e = (1, 0, 0, \ldots, 0)^T$
$b_1, b_2, \ldots, b_{n-1}$- первые столбцы соответствующих матриц $B_1, B_2, \ldots, B_{n-1}$

3 Результаты и вывод

3.1 Входные данные

0.6444 0.0000 -0.1683 0.1184 0.1973
-0.0395 0.4208 0.0000 -0.0802 0.0263
0.0132 -0.1184 0.7627 0.0145 0.0460
0.0395 0.0000 -0.0960 0.7627 0.0000
0.0263 -0.0395 0.1907 -0.0158 0.5523

3.2 Выходные данные

```
Ищем собственные вектор матрицы
Исходная матрица:
[[ 0.4965221  -0.02976049 -0.10906373  0.13191428  0.09195098]
 [-0.02976049  0.18575662 -0.05029722 -0.06272879 -0.0018678 ]
 [-0.10906373 -0.05029722  0.59823034 -0.06163865  0.17564755]
 [ 0.13191428 -0.06272879 -0.06163865  0.59248754 -0.02931901]
 [ 0.09195098 -0.0018678   0.17564755 -0.02931901  0.34390336]]
Метод Леверье
Коэфициенты характеристического многочлена
 [ 2.21689996 -1.82259145  0.68304846 -0.11469321  0.0069546 ]
q_1-Sp(A) =  1.92392075075 q_n-det(A) =  0.00695459981768
Метод Фаддеева
Коэфициенты характеристического многочлена
 [ 2.21689996 -1.82259145  0.68304846 -0.11469321  0.0069546 ]
q_1-Sp(A) =  0.0 q_n-det(A) =  2.60208521397e-18
Собственное значение:  0.780834944861
Собственный вектор:  [ 0.00382651 -0.00022833 -0.00553064  0.00483712 -0.00174166]
Невязка:  [ -6.33174069e-17   5.42101086e-19   6.07153217e-18  -9.10729825e-18
  -4.55364912e-18]
Норма невязки:  6.44199502585e-17
```

3.3 Вывод

Как видно из полученных результатов, расхождение у коэффициентов минимального многочлена, полученных этими способами начинаются только в -16 порядке, так они построены на одинаковых рассуждениях, а само расхождение появляется только в результате вычислительных погрешностей. Модификация метода Леверрье Фаддеевым позволяет получить не только минимальный многочлен, но и построить соответствующие собственные вектора, с точностью выше, чем у метода Данилевского и ниже, чем у метода Крылова.

4 Листинг кода

```
import numpy as np
import numpy.linalg as linalg

def leberie(matr_A):
    A = matr_A.copy()
    n = len(A)
    s = np.zeros(n + 1)
    for i in range(1, n + 1):
        s[i] = sum(A[j, j] for j in range(n))
        A = np.dot(A, matr_A)
    p = np.zeros(n + 1)
    for i in range(1, n + 1):
        p[i] = (s[i] - sum(s[j]*p[i-j] for j in range(i))) / i
    print("Коэфициенты характеристического многочлена\n", p[1:])
    print("q_1-Sp(A) = ", abs(A.trace() - p[1]), "q_n-det(A) = ", abs(linalg.det(A) - p[n]))
    return p[1:]

def faddeev(matr_A):
    A = matr_A.copy()
    n = len(A)
    q = np.zeros(n + 1)
    resB = np.eye(n)
    E = np.eye(n)
    for i in range(1, n + 1):
        q[i] = sum(A[i, i] for i in range(n)) / i
        B = A - q[i]*E
        resB[:,i-1] = B[:,0].copy()
        A = np.dot(matr_A, B)
    print("Коэфициенты характеристического многочлена\n", q[1:])
    print("q_1-Sp(A) = ", abs(matr_A.trace() - q[1]), "q_n-det(A) = ", abs(linalg.det(matr_A) - q[n]
    p = [1] + list(q[1:] * -1)
    eigvals = np.roots(p)
    v = (eigvals[0] ** (n - 1))*E[:, 0]
    for i in range(n - 1):
        v += (eigvals[0] ** i) * resB[:,(n - i - 2)]
    return v, eigvals[0]

file = open("matrix", "r")  # Чтение файла
A, b = [], []
for line in file:
    A.append([float(el) for el in line.split()[:-1]])
    b.append(float(line.split().pop()))
A = np.array(A)
A = np.dot(A, A.transpose())
b = np.array(b)
print("Ищем собственные вектор матрицы")
print("Исходная матрица:")
print(A)
print("Метод Леверье")
leberie(A)
print("Метод Фаддеева")
ans = faddeev(A)
print("Собственное значение: ", ans[1])
print("Собственный вектор: ", ans[0])
print("Невязка: ", np.dot(A, ans[0]) - ans[0] * ans[1])
print("Норма невязки: ", linalg.norm(np.dot(A, ans[0]) - ans[0] * ans[1]))
```